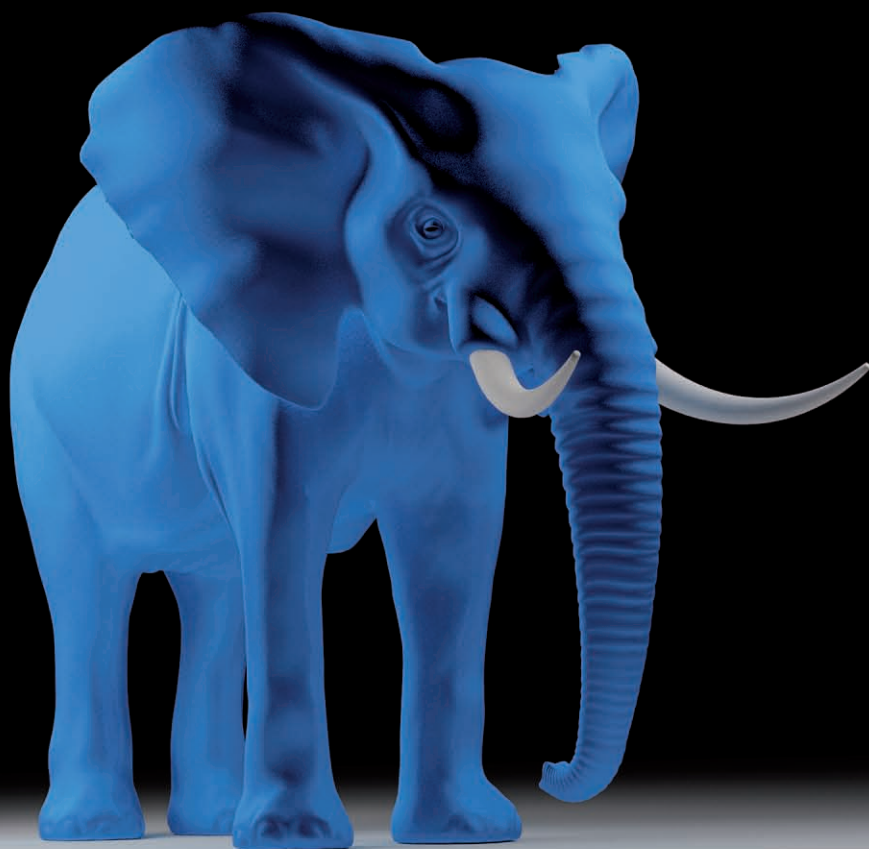


CARLOS PAMPULIM CALDEIRA

# PostgreSQL

## Guia fundamental



EDIÇÕES SÍLABO



# PostgreSQL

## Guia Fundamental

CARLOS PAMPULIM CALDEIRA

*EDIÇÕES SÍLABO*

É expressamente proibido reproduzir, no todo ou em parte, sob qualquer forma ou meio, **NOMEADAMENTE FOTOCÓPIA**, esta obra. As transgressões serão passíveis das penalizações previstas na legislação em vigor.

Visite a Sílabo na rede

[www.silabo.pt](http://www.silabo.pt)

Editor: Manuel Robalo

FICHA TÉCNICA:

Título: PostgreSQL – Guia Fundamental

Autor: Carlos Pampulim Caldeira

© Edições Sílabo, Lda.

Capa: Pedro Mota

1ª Edição – Lisboa, Março de 2015

Impressão e acabamentos: Cafilesa – Soluções Gráficas, Lda.

Depósito Legal: 388914/15

ISBN: 978-972-618-795-0

EDIÇÕES SÍLABO, LDA.

R. Cidade de Manchester, 2

1170-100 Lisboa

Tel.: 218130345

Fax: 218166719

e-mail: [silabo@silabo.pt](mailto:silabo@silabo.pt)

[www.silabo.pt](http://www.silabo.pt)

# Índice

<b>Índice de figuras e quadros</b>	11
<b>Convenções usadas neste manual</b>	15

## Capítulo 1

---

### **Introdução**

1.1. Breve história do PostgreSQL	18
1.2. O que é uma base de dados?	19
1.3. A organização e gestão dos dados	19
1.4. Representação de dados numa base de dados	20
1.5. PostgreSQL – algumas características	22

## Capítulo 2

---

### **O Ambiente SQL**

2.1. Concepção de bases de dados em PostgreSQL	28
2.2. A questão das maiúsculas e minúsculas em PostgreSQL	29
2.2.1. Linguagem corrente e DDL	30
2.2.2. A comparação de dados e as letras maiúsculas e minúsculas	32
2.3. Denominação formal	35
2.4. Criação de uma tabela em SQL	35
2.5. Tipos de dados	37
2.5.1. Tipos de dados textuais	37
2.5.2. Grandes cadeias alfanuméricas	40

2.5.3. Tipos de dados numéricos	41
2.5.4. Conversão entre cadeiras alfanuméricas e valores numéricos	44
2.5.5. Sequências	44
2.5.6. Arrays	45
2.5.7. Dados binários (imagens)	53
2.5.8. Tipos lógicos	56
2.5.9. Tipos temporais (datas e horas)	59
2.6. Índices	68

## Capítulo 3

---

### **Pesquisa de Informação em SQL**

3.1. Terminologia	73
3.2. Pesquisa de dados	74
3.3. Queries de primeiro nível	76
3.4. A ordenação dos resultados	84
3.4.1. Ordenamento simples	84
3.4.2. Ordenamento composto	86
3.5. Associação entre tabelas	87
3.6. Funções de agregação	89
3.6.1. A cláusula GROUP BY	91
3.6.2. Restrições sobre grupos	92
3.7. Queries aninhadas	93
3.7.1. Operadores em subqueries	96
3.8. UNION, INTERSECT e EXCEPT	99
3.9. Afinação do desempenho das queries	103
3.9.1. EXPLAIN e EXPLAIN ANALYZE	104
3.9.2. Como escrever melhores queries	108

## Capítulo 4

---

### **Gestão da Qualidade dos Dados**

4.1. Chaves em SQL	120
4.1.1. Chave primária	120
4.1.2. Como declarar as chaves	120

4.1.3. Chave única	121
4.1.4. Como se escolhe uma chave primária	122
4.1.5. Chaves primárias e colunas únicas	124
4.1.6. Duas tabelas com a mesma chave primária?	124
4.1.7. A questão das maiúsculas e minúsculas nas chaves	125
4.2. Preenchimento obrigatório	127
4.3. Restrições nos valores	128
4.4. Integridade referencial	131
4.4.1. Representação gráfica da chave estrangeira	134
4.4.2. O PostgreSQL e as subcláusulas da integridade referencial	134
4.5. Mais sobre CHECK	135
4.6. Alteração das regras de integridade	137
4.6.1. Utilização dos nomes das restrições	138
4.6.2. Alteração de restrições sobre tabelas	138
4.6.3. Remoção de restrições	139
4.6.4. Adição de restrições	139
4.7. Triggers	139
4.7.1. Estrutura dos gatilhos	140
4.8. Rules	145
4.9. Módulo Itree	146
4.9.1. Índices	149
4.9.2. Exemplo de aplicação	149
4.9.3. Common Table Expressions (With Queries)	154

## Capítulo 5

---

### **Desenho Físico da Base de Dados**

5.1. Regras para o desenho de tabelas	161
5.1.1. Normalização de tabelas	165
5.1.2. Orientação para a criação de índices	173

## Capítulo 6

---

### **Transações em SQL**

6.1. Causas para a inconsistência dos dados	178
6.1.1. Processamento em série	178
6.2. Atomicidade	180
6.2.1. Propriedades ACID das transações	182
6.3. A arquitectura dos sistemas e as transações	182
6.3.1. Controlo de acesso concorrential	183
6.4. Transações e níveis de isolamento	189
6.4.1. Nível de isolamento Read Committed	191
6.4.2. Nível de isolamento Repeatable read	193
6.4.3. Nível de isolamento Serializable	194
6.5. Views	198
6.5.1. Criação de view	199
6.5.2. View horizontal	202
6.5.3. View vertical	203
6.5.4. Vantagens e desvantagens das views	203
6.5.5. Vistas materializadas	205

## Capítulo 7

---

### **Gestão de Utilizadores**

7.1. Atribuição de privilégios	210
7.2. O <papel>	211
7.3. Atributos dos papéis	212
7.4. Agrupamento de papéis	213
7.5. Privilégios em PostgreSQL	214
7.6. Matriz de autoridade	216
7.6.1. Matriz de autoridade de nível 1	216
7.6.2. Matriz de autoridade de nível 2	217



## **Anexos**

8.1. CREATE TABLE	219
8.2. ALTER TABLE	221
8.3. DROP TABLE	224
8.4. GRANT	225
8.5. REVOKE	227
8.6. CREATE ROLE	229
8.7. DROP ROLE	231
8.8. ALTER ROLE	232
8.9. CREATE SEQUENCE	233
8.10. DROP SEQUENCE	234
8.11. ALTER SEQUENCE	235
8.12. CREATE MATERIALIZED VIEW	236
8.13. ALTER VIEW	237
8.14. CREATE RULE	238
8.15. SELECT	240
8.16. Base de dados do caso prático de SELECT	242
8.16.1. Conteúdo das tabelas	242
8.16.2. Comandos de criação das tabelas	244
8.17. Tabelas com muitas linhas	247
8.17.1. Tabela Eventos	247
8.17.2. Tabela Descrição dos Eventos	247
8.18. Operadores e funções sobre arrays	248
8.19. Elementos de escantilhão para formatação de datas e horas	251
8.20. Elementos de escantilhão para formatação numérica	253
<b>Índice remissivo</b>	<b>255</b>



# Índice de figuras e quadros

## ■ Figuras

Figura 1.1. Esquema da base de dados Cinema	21
Figura 1.2. A tabela Filme	21
Figura 1.3. A tabela Filme com valor NULL (nulo) na coluna Duração	22
Figura 2.1. Estrutura organizacional do PostgreSQL	27
Figura 2.2. Query com concatenação	39
Figura 2.3. Aparente duplicação numa chave	40
Figura 2.4. Array com três linhas	49
Figura 2.5. Tabela fruta	54
Figura 2.6. Resultado de LO_EXPORT num comando de SQL	55
Figura 2.7. Mini gestão de stocks com valores lógicos	57
Figura 2.8. Operadores de adição, subtração e sobreposição entre datas	62
Figura 3.1. Base de dados para os exemplos do comando SELECT	76
Figura 3.2. Forma gráfica de EXPLAIN	105
Figura 3.3. A utilização de cores progressivamente mais fortes (amarelo, laranja e vermelho) indica um caso mais sério de perda de performance da query	106
Figura 3.4. Tabela de estatísticas da ferramenta de Hubert Lubaczewski	106
Figura 3.5. Melhoria de pontos “negros” no desempenho de uma query	107
Figura 3.6. Tabela com estatísticas actualizadas	107
Figura 3.7. As primeiras linhas da tabela Eventos	111
Figura 3.8. Resultado da query do Exemplo 3.20	116
Figura 4.1. Representação gráfica das chaves primária e únicas	122
Figura 4.2. Excerto de base de dados	124
Figura 4.3. Valor repetido numa coluna UNIQUE	125
Figura 4.4. Tabela com valores “duplicados” na chave primária	126
Figura 4.5. Aplicação da função UPPER para contornar a distinção entre maiúsculas e minúsculas numa chave única	127
Figura 4.6. Ligação entre as tabelas Sistema de Cor e Filme	134

Figura 4.7. Tabela de auditoria com duas linhas	145
Figura 4.8. Hierarquia da colecção de comboios	150
Figura 4.9. Plano de execução de uma query recursiva	158
Figura 4.10. Resultado da query recursiva “a partir de 2 locais”	159
Figura 5.1. Exemplo de uma tabela de uma base de dados relacional	161
Figura 5.2. Tabela Filme com uma chave primária composta (CP) por duas colunas	162
Figura 5.3. Tabela Género com uma chave primária abstracta: GéneroID	163
Figura 5.4. Tabela Género num formato normalizado	163
Figura 5.5. Esquema Cinema com representação de chaves estrangeiras	164
Figura 5.6. Tabela com filme com três realizadores	166
Figura 5.7. Tabela com um grupo de colunas repetidas	166
Figura 5.8. Tabela Realizador do Filme	167
Figura 5.9. Sub esquema cinema com as tabelas Filme e Realizador do Filme	167
Figura 5.10. Sub esquema cinema com as tabelas Realizador, Filme e Realizador do Filme com chaves estrangeiras	168
Figura 5.11. Aplicação da integridade referencial	169
Figura 5.12. Exemplo de uma caixa de combinação	169
Figura 6.1. Tabelas Empregado e Actor	176
Figura 6.2. Diagrama de estado para uma transação	178
Figura 6.3. Dupla reserva sobre o mesmo lugar	180
Figura 6.4. O problema da alteração perdida	184
Figura 6.5. Exemplo de uma alteração perdida na actualização de dados na turma x	185
Figura 6.6. Exemplo do problema da leitura suja	186
Figura 7.1. Exemplo de matriz de nível 1	217
Figura 7.2. Matriz de nível 2 para o papel “recursos humanos”	217
Figura 7.3. Os privilégios dos roles todos numa única folha	218

## ■ Quadros

Quadro 1.1. Limites máximos do PostgreSQL	18
Quadro 2.1. Exemplo de extensões instaladas no PostgreSQL	34
Quadro 2.2. Tipos de dados numéricos	41
Quadro 2.3. Funções de conversão entre valores textuais e numéricos	44
Quadro 2.4. Operadores sobre marcas temporais	65
Quadro 2.5. Funções de formatação de datas e horas	65
Quadro 2.6. Formatos para entrada de datas	66
Quadro 2.7. Formatos para entrada de valores horários	67
Quadro 2.8. Formatos para entrada de fuso horário (considerando hora de Verão em Portugal)	67

Quadro 3.1. Comandos DML em SQL	71
Quadro 3.2. Operadores de comparação e lógicos	72
Quadro 3.3. Resultado do Exemplo 3.1	77
Quadro 3.4. Resultado do Exemplo 3.2	77
Quadro 3.5. Resultado do Exemplo 3.3 (a)	78
Quadro 3.6. Resultado do Exemplo 3.3 (b)	78
Quadro 3.7. Resultado do Exemplo 3.4 (a)	79
Quadro 3.8. Resultado do Exemplo 3.4 (b)	79
Quadro 3.9. Resultado do Exemplo 3.5	80
Quadro 3.10. Operadores de comparação em PostgreSQL	81
Quadro 3.11. Resultado do Exemplo 3.6	81
Quadro 3.12. Resultado do Exemplo 3.7	82
Quadro 3.13. Resultado do Exemplo 3.9	83
Quadro 3.14. Resultado do caso 1 do Exemplo 3.10	83
Quadro 3.15. Resultado do caso 2 do Exemplo 3.10	84
Quadro 3.16. Ordenação pelo nome do Actor – caso 1	85
Quadro 3.17. Ordenação pela denominação da expressão	86
Quadro 3.18. Listagem com duas chaves de ordenamento	87
Quadro 4.1. Dados de uma colecção de comboios	150
Quadro 4.2. Organização hierárquica das locomotivas	150
Quadro 4.3. Rotas com etiqueta de vapor	151
Quadro 4.4. Rotas com locomotiva a vapor mas sem fotografia (pesquisa em texto)	151
Quadro 4.5. Tabela voos	156
Quadro 5.1. Lista de realizadores para um filme	168
Quadro 5.2. Tabela que não obedece à 2FN	170
Quadro 5.3. Normalização de tabela para a 2FN	171
Quadro 5.4. Tabela não cumpre com a 3FN	171
Quadro 5.5. Normalização da tabela Empregado para a 3FN	172
Quadro 6.1. Movimentação de dinheiro e a atomicidade transaccional	181
Quadro 6.2. Análise de dados com resultado errado	187
Quadro 6.3. Níveis de isolamento e transações	190
Quadro 6.4. Percurso de duas transações concorrentes entre os tempos t1 a t6	198
Quadro 6.5. Resultado final das transações T1 e T2 na escala de enfermeiros	198
Quadro 6.6. Algumas vantagens e desvantagens das views	204
Quadro 7.1. Permissões de um papel sobre objectos da base de dados	215



# Convenções usadas neste manual

<b>Comandos em SQL</b>	SELECT	Letra maiúscula
<b>Nome dos objectos da base de dados</b>	Actor	Combinação de letras maiúsculas e minúsculas
<b>Aspas (simples ou duplas)</b>	“Nome do Actor”	Denominação completa dos objectos da base de dados
<b>Parêntesis recto</b>	[NOT NULL]	Cláusula   Comando opcional
<b>Chavetas</b>	{ON   OFF}	Deve introduzir-se obrigatoriamente uma das condições separadas pelo “ ”. As chavetas e o “ ” não são para incluir no comando
<b>Itálico</b>	<i>N</i>	Parâmetro que se destina a ser substituído por um valor
<b>Barra vertical</b>		Deve digitar-se obrigatoriamente um dos dois argumentos





# Capítulo 1

# Introdução

Este livro é sobre um sistema de bases de dados que tem passado despercebido à maior parte dos utilizadores deste tipo de aplicações. Enquanto quase toda a gente já ouviu falar do Oracle, do SQL Server ou até do MySQL poucas são as que reconhecem o PostgreSQL. No entanto, esta base de dados já existe há muitos anos e, em alguns casos, até antes de algumas das mais populares.

Em linhas gerais pode dizer-se que o PostgreSQL é um sistema de gestão de bases de dados relacional (SGBDR) semelhante a todos os outros.

Mas, uma análise mais minuciosa dá-nos a conhecer um sistema que vai para além do mínimo exigível a um SGBDR, podendo mesmo ser considerado como uma solução de gestão de dados de alto rendimento e, que tem ainda a vantagem de funcionar em diversas plataformas informáticas, *e.g.*, Windows, Linux, Unix, Mac OS.

O PostgreSQL não tem nenhum complexo de inferioridade e rivaliza, em todos os aspectos, com os pesos-pesados desta área como sejam o Oracle ou o SQL Server.

À versão de base podem ser acrescentadas diversas funcionalidades adicionais como sejam:

- Sistema de informação geográfica.
- Processamento distribuído.
- Clustering de sistemas e de bases de dados.

Este sistema resistiu à prova do tempo e da evolução tecnológica, tendo já provado em inúmeras situações que está apto a resolver qualquer problema na área da gestão de dados.

E, o melhor mesmo é que NÃO CUSTA NADA. É um projecto open source: livre para descarregar e livre para utilizar.

O PostgreSQL é praticamente 100% compatível com o ISO SQL e, corre em diferentes sistemas operativos. Além disso, é muito fácil de instalar sem grandes complicações. Os limites máximos do sistema em diversos aspectos são impressionantes como se pode ver no Quadro 1.1.

**Quadro 1.1. Limites máximos do PostgreSQL**

Objecto	Limite
Tamanho máximo da base de dados	Ilimitado
Tamanho máximo de uma tabela	32 Terabytes
Comprimento máximo de uma linha	1,6 Terabytes
Tamanho máximo de uma coluna	1 Gigabyte
Número máximo de linhas numa tabela	Ilimitado
Número máximo de colunas numa tabela	250-1600 <sup>1</sup>
Número máximo de índices por tabela	Ilimitado

<sup>1</sup> Dependendo dos tipos de dados das colunas.

---

## 1.1. Breve história do PostgreSQL

O PostgreSQL foi desenvolvido em 1986 pelo professor Michael Stonebraker da Universidade UCB, com a denominação original de Postgres, como o sucedâneo a outra aplicação chamada de Ingres, e daí o “Post” “gres” no sentido de “após/depois do Ingres”.

Stonebraker e os seus alunos desenvolveram o Postgres de 1986 a 1994 acrescentando-lhe muitas características hoje imprescindíveis e fundamentais num SGBDR de alto nível.

Dois desses alunos (Andrew Yu e Jolly Chen) substituíram em 1995 a linguagem de query do Postgres (o Postquel) pelo SQL, e baptizaram o novo sistema como Postgres95.

A partir de 1996 o Postgre95 emancipou-se do meio académico e foi perfilhado pela comunidade open source onde um grupo de especialistas se dedicou ao traba-

lho de transformar uma jóia em bruta numa refinada peça de joalheria. Todo esse trabalho resultou num sistema estável, com altos desempenhos e apetrechado com tudo o que é de mais moderno no mundo das bases de dados.

---

## 1.2. O que é uma base de dados?

Uma base de dados, no sentido estrito de “qualquer coisa que informa”, é composta por **tabelas**, que por sua vez estão estruturadas em **linhas** e **colunas**, e cada linha tem uma **identidade própria** na tabela.

Cada linha é interceptada por pelo menos uma coluna, e cada um desses pontos de intersecção é uma **peça de informação**.

Uma base de dados é do tipo **relacional** quando tem origem no Modelo de Dados Relacional. Ou seja, como um prédio tem plantas e é construído com suporte nesse planeamento, também uma base de dados tem que ser pensada e conceptualizada. E é para isso que servem os modelos de dados: são as plantas das bases de dados. Na prática o qualificativo relacional significa que é possível associar os dados localizados em tabelas distintas através de colunas que lhes sejam comuns.

É esta capacidade de relacionar os dados contidos em tabelas diferentes que é o ponto forte das bases de dados relacionais. Aliás, é essa característica que permite transformar **dados** (aquilo que está em cada uma das tabelas) em **informação**. A informação nasce precisamente dessa faculdade de criar informação ao ligar as tabelas entre si.

Como se pode ver na Figura 1.1 (pág. 21) é possível associar os actores que participaram num determinado filme através da tabela Actores do Filme que tem colunas em comum com as tabelas Filme e Actor.

---

## 1.3. A organização e gestão dos dados

A informação numa base de dados relacional está guardada em tabelas que são constituídas por linhas e colunas. As tabelas organizam-se numa base de dados de acordo com um determinado esquema. Cada tabela é um objecto bem definido, com um propósito concreto, e com uma denominação única no esquema.

Uma tabela é normalmente constituída por muitas colunas e linhas mas há casos particulares em que pode existir apenas uma única coluna; ou o inverso também pode ser verdadeiro: uma tabela com várias colunas mas constituída unicamente por uma linha, a qual por sua vez pode estar vazia ou conter dados.

Os dados nas tabelas estão contidos nas linhas e são as colunas que dão contexto a esses dados. Para um determinado conjunto de colunas os valores (nessas colunas) são únicos em cada uma das linhas. É este facto que permite distinguir as linhas entre si. Esse conjunto identificador denomina-se chave primária (CP). O conjunto identificador pode ser singular ou plural, no primeiro caso é formado por uma só coluna – pelo que se diz chave primária simples – no segundo por múltiplas colunas em simultâneo – e então denomina-se chave primária composta. No entanto, quer seja simples ou composta uma tabela tem sempre e unicamente uma chave primária.

Uma base de dados relacional precisa de uma interface especial com o utilizador. Essa interface que possibilita quer a simples interacção com os dados, ou as mais complexas tarefas de gestão da informação e da estrutura da base de dados é o Sistema de Gestão de Base de Dados Relacional. Quando se fala do Oracle, PostgreSQL ou SQL Server, por exemplo, está-se a referir a SGBDRs. Todos os modernos e mais utilizados SGBDRs utilizam o SQL (Structured Query Language) como o “caminho” de acesso à informação.

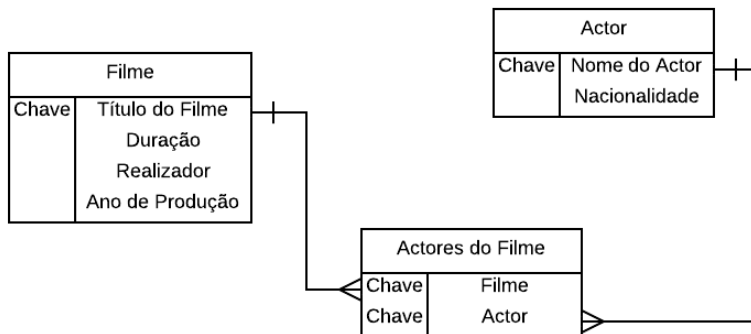
Um SGBDR além de ser composto por dados (aquilo que é mais evidente para os utilizadores sem conhecimentos técnicos) engloba uma série de componentes que permitem e facilitam a entrada e manipulação dos dados, ou dito de outro modo, um sistema de gestão de base de dados é um conjunto de software que gere e controla o acesso à informação propriamente dita.

---

## 1.4. Representação de dados numa base de dados

Uma tabela típica numa base de dados relacional contém dados sobre um único tema, ou objecto-informativo, e tem uma denominação unívoca no esquema em que está incluída. Já um esquema é uma representação de um processo de negócio. Por exemplo, o esquema da Figura 1.1 denomina-se Cinema e é composto pelas tabelas Filme, Actor e Actores do Filme; neste esquema só pode haver uma tabela Actor.

Figura 1.1. Esquema da base de dados Cinema



Uma tabela tem que ser constituída por pelo menos uma coluna, e uma coluna é um conjunto de valores de um certo tipo de dados: nomes, números de telefone, datas, entre muitas outras possibilidades. Numa tabela não podem coexistir colunas com o mesmo nome. Por outro lado, tanto quanto possível essa denominação deve explicitamente indicar qual o tipo de dados tratados na coluna.

Figura 1.2. A tabela Filme

Título do Filme	Duração	Realizador	Ano de Produção
King Kong	100	Merian C. Cooper, Ernest B. Schoedsack	1933
Gone with the Wind	238	Victor Fleming	1939
King Kong	187	Peter Jackson	2005

← **linha**

↑  
**coluna**

A tabela Filme (Figura 1.2) tem quatro colunas: uma para o nome do filme (Título do Filme), outra para a duração em minutos (Duração), uma outra para registar o nome do realizador (Realizador) e, finalmente, uma para o ano de produção do filme (Ano de Produção).

Na Figura 1.2, tabela Filme, o título do filme "Gone with the Wind" é uma peça de informação atômica, ou simplesmente, uma peça de informação. Por sua vez cada linha é um objecto distinto dentro da tabela, ou dito de outro modo, cada linha é

irrepetível. Ou seja, a linha da tabela Filme em que o título é "King Kong" e o ano de produção "1933" é um conjunto único de dados.

Regra geral a intercepção de uma coluna com uma linha origina sempre um valor, mas pode acontecer que ele não exista, como no caso da coluna Duração no filme "Apocalypse Now", e nesse caso a coluna tem o valor de NULL (Figura 1.3). O valor NULL não é zero, nem um espaço em branco, é pura e simplesmente a ausência de um valor. Numa tabela sempre que não se conhece o valor para uma coluna não se deve usar nenhum dos truques clássicos da informática como, por exemplo, preencher com "-", ou "\_". É sempre preferível deixar o campo a NULL.

**Figura 1.3. A tabela Filme com valor NULL (nulo) na coluna Duração**

Título do Filme	Duração	Realizador	Ano de Produção
King Kong	100	Merian C. Cooper, Ernest B. Schoedsack	1933
Gone with the Wind	238	Victor Fleming	1939
King Kong	187	Peter Jackson	2005
Apocalypse Now		Francis Ford Coppola	1979

---

## 1.5. PostgreSQL – algumas características

O PostgreSQL é considerado o sistema de gestão de base de dados relacional open source mais avançado do mundo. Passe o exagero, este sistema tem de facto uma estrutura suportada em alguns atributos que lhe dão um certo número de vantagens:

- Apesar de ser um SGBDR com alto nível de performance é todavia bastante amigável na utilização do dia-a-dia, e de fácil administração.
- Utiliza a linguagem standard SQL.
- É um sistema multithreaded, ou de junção múltipla, pelo que aceita a ligação de numerosos clientes e, por sua vez, cada cliente pode utilizar várias bases de dados em simultâneo. Um servidor de PostgreSQL tem a possibilidade de funcionar nos sistemas operativos mais conhecidos, Windows, Linux, Unix e Mac OS, e aceita ligações a partir de sistemas operativos externos pois suporta os protocolos ODBC, JDBC e .NET. A utilização destes protocolos permite que os mais diversos tipos de clientes tenham acesso às bases de dados

em PostgreSQL. Assim, é possível, por exemplo, aceder de um modo transparente à informação através de uma folha de cálculo.

- O PostgreSQL tem altos níveis de conectividade podendo ser acedido a partir de qualquer nó da Internet. Para isso este sistema relacional possui apertados controlos de acesso: para além das credenciais de acesso associadas a cada utilizador, tem ainda regras de acesso aos dados que permitem que, *e.g.*, um conjunto de informação possa ser visto por dois utilizadores mas que só um deles os possa modificar. Já em termos de segurança de conexão emprega ligações encriptadas através do protocolo SSL (*Secure Sockets Layer*).
- Este SGBDR é um projecto open source desenvolvido desde 1996 pelo *The PostgreSQL Global Development Group* (<http://www.postgresql.org>).

**Carlos Pampulim Caldeira** é Professor Auxiliar no Departamento de Informática da Universidade de Évora ([www.di.uevora.pt/~ccaldeira](http://www.di.uevora.pt/~ccaldeira)), tem um Doutoramento em Sistemas de Informação pela Universidade Técnica de Lisboa, é membro da *Association for Computing Machinery*, da *European Association of Software Science and Technology* e do *Independent Oracle Users Group*. É pioneiro na utilização de sistemas de bases de dados em Portugal, desenvolveu e coordena diversas disciplinas nas áreas das bases de dados, *data warehousing* e *business intelligence*. Desenvolve desde 2000 investigação na área da Ecologia da Informação. Coordenou o desenvolvimento de vários projectos de bases de dados na Administração Pública e em empresas privadas.

O PostgreSQL é o sistema de gestão de base de dados relacional *open source* mais avançado da actualidade. Pelo facto de ser de acesso livre, o PostgreSQL é uma ferramenta de eleição quer para as pequenas e médias empresas quer para a Administração Pública Central e Local. A opção por este tipo de ferramenta pode ser um factor determinante na diminuição da despesa com *software* aplicacional.

O seu desenvolvimento é assegurado por um grupo de profissionais altamente qualificados, muitos dos quais já passaram pelas companhias mais conhecidas no universo dos sistemas relacionais comerciais. Dessa experiência nasceu um produto rápido, robusto e com limites máximos, em termos de dados, de metadados e de definição das estruturas de dados, absolutamente brilhantes.

O PostgreSQL, devido às suas características topo de gama, consegue suportar um leque muito diversificado de solicitações: desde sistemas pessoais até complexos sistemas em *cluster*/distribuídos. Desde a pequena indústria ou comércio até uma organização com múltiplas localizações. A sua observância em relação ao ISO SQL é uma garantia de qualidade e de fiabilidade do sistema em si mesmo e das aplicações associadas.

Quer se trate de um iniciado ou de um profissional com experiência em bases de dados, todos encontrarão neste livro material específico que lhes permitirá entrar no mundo das bases de dados relacionais através desta ferramenta tão extraordinária que é o PostgreSQL.



# PostgreSQL

## Guia fundamental

499

